

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method of processing data, ~~using an object-oriented programming language~~, comprising:

defining compiling a definition of a class which supports an options, each option defined in the class or through a class inheritance hierarchy, and each option having an option value associated therewith ~~data structure having,~~

wherein in instances of the class, references to the options values are referenceable without allocation of memory space for the associated ~~full~~ option values when ~~the instances of the class is are~~ created, ~~the option data structure including a type description of the option values; and~~

wherein compiling the definition comprises

generating a class describing data structure, wherein the class describing data structure is associated with at least one option describing data structure, the at least one option describing data structure comprising a type description corresponding to each option defined in the class or through the class inheritance hierarchy; and

compiling an operation on an value associated with a selected option ~~value~~ in an instance of the class, ~~using the type description in the option data structure~~

wherein compiling the operation comprises searching the at least one option describing data structure associated with the class describing data

structure to locate the type description corresponding to the selected option, and
using the type description to type check the operation.

2. (Currently Amended) A method as claimed in claim 1 wherein the at least one
option describing data structure identifies change handler code that is executed when
an option value changes.

3. (Currently Amended) A method as claimed in claim 2 wherein the selected
option is defined in a plurality of classes within a class inheritance hierarchy, with each
of the plurality of classes identifying change handler code corresponding to the selected
for one option, and further comprising, when the option value associated with the
selected option changes, executing is defined in different classes within a class-
inheritance hierarchy and the change handler code from identified by each of the
plurality of classes is executed when the in which the selected option value changes is
defined.

4. (Currently Amended) A method as claimed in claim 1 wherein the at least one
option describing data structure ~~includes further comprises~~ a default value for each
option defined in the class or through the class inheritance hierarchy, the method further
comprising, in a get operation to a selected option in an instance of the class, ~~if an~~ when
the option value which applies to associated with the instances selected option has
been set, getting the set option value and, ~~if no~~ when the option value which applies-

associated with the selected option has not been set, getting the default value ~~for the~~
class.

5. (Canceled).

6. (Currently Amended) A method as claimed in claim 1, wherein the at least
one option describing data structure comprises an option binding object for each option
defined in the class or through the class inheritance hierarchy, further comprising:

notifying ~~objects~~ a selected instance of the class of a change in an option value
through a change handler identified by ~~an~~ the option binding object, the option binding
object being located by first searching a mapping data structure for a previously
computed mapping to the option binding object and, if no mapping was previously
computed, by then computing the mapping to the option binding object and storing the
mapping in the mapping data structure.

7. (Currently Amended) A method as claimed in claim 1 wherein the instance of
the class is associated with a listing option-data structure, the listing data structure
having option items comprising option values, the option items being arranged in
~~comprises a linked list of option items having option values~~.

8. (Currently Amended) A method as claimed in claim 1 wherein a nonlocal
option value applies to other ~~objects~~ instances of the class in a nonlocal option
hierarchy.

9. (Original) A method as claimed in claim 8 wherein the nonlocal option hierarchy is a graphical hierarchy.

10. (Currently Amended) A method as claimed in claim 1 wherein the class ~~which supports the option data structure includes defined fields, each field having a field value associated therewith, wherein instances of the class are allocated memory space for the associated field value when the instance of the class is created to support values in preallocated memory space.~~

11. (Currently Amended) A method as claimed in claim 1, wherein the operation is to set the option value to a second value, the second value having a declared type, and wherein the type checking the operation comprises description is used to checking the declared type of a value to be set in a set operation against the type description corresponding to the selected option.

12. (Currently Amended) A method as claimed in claim 1 wherein the operation is one that is to be performed using a selected option value resulting from a get operation, and wherein type checking the operation comprises description is used to checking the legality of an the operation to be performed on a using the selected option value obtained in a get operation.

13. (Currently Amended) A data processing system ~~using an object oriented programming language~~ comprising:

a memory; and

a processor coupled to the memory and configured

for compiling a definition of a class which supports an options, each option defined in the class or through a class inheritance hierarchy, and each option having option values associated therewith ~~data structure having,~~

wherein in instances of the class, references to the options values are referenceable without allocation of memory space for the full associated option values when the instances of the class is are created, ~~the option data structure including a type description of the option values;~~ and

wherein compiling the definition comprises

generating a class describing data structure, wherein the class describing data structure is associated with at least one option describing data structure, the at least one option describing data structure comprising a type description corresponding to each option defined in the class or through the class inheritance hierarchy, and

a compiler which, when for compiling an operation on an option value associated with a selected option in an instance of the class, wherein compiling the operation comprises searching the option describing data structure associated with the class describing data structure to locate the type description

corresponding to the selected option, and using ~~uses~~ the type description to type
check the operation ~~of the option value in the option data structure.~~

14. (Currently Amended) A system as claimed in claim 13 wherein the at least
one option describing data structure identifies change handler code that is executed
when an option value changes.

15. (Currently Amended) A system as claimed in claim 14 wherein the selected
~~change handler code for one option~~ is defined in ~~different~~ a plurality of classes within a
class inheritance hierarchy, with each of the plurality of classes identifying ~~and the~~
~~change handler code~~ corresponding to the selected option, and wherein when the
option value associated with the selected option changes, executing the change handler
code from each identified by each of the plurality of classes in which the selected option
is defined ~~class is executed when the option value changes.~~

16. (Currently Amended) A system as claimed in claim 13 wherein the at least
one option describing data structure ~~includes~~ further comprises a default value
corresponding to each option defined in the class or through the class inheritance
hierarchy, which and wherein the default value is obtained when no option value has
been set ~~in an applicable instance object~~ for the corresponding option.

17. (Canceled).

18. (Currently Amended) A system as claimed in claim 13, wherein the at least one option describing data structure comprises an option binding object associated with each option defined in the class or through the class inheritance hierarchy, each option having an option name, further comprising ~~change handlers which notify objects of a change in an option value and~~ a mapping data structure which maps an selected option name and class to an the associated option binding object, wherein the associated option binding object ~~which identifies a change handlers that notify instances of the class of a change in the selected option value.~~

19. (Currently Amended) A system as claimed in claim 13 wherein the instance of the class is associated with a ~~option-listing~~ data structure, the listing data structure comprising ~~comprises~~ a linked list of option items having option values.

20. (Currently Amended) A system as claimed in claim 13 wherein a nonlocal option value applies to other ~~objects~~ instances of the class in a nonlocal option hierarchy.

21. (Original) A system as claimed in claim 20 wherein the nonlocal option hierarchy is a graphical hierarchy.

22. (Currently Amended) A system as claimed in claim 13 wherein the class ~~which supports the option data structure includes defined fields,~~ each field having a field value associated therewith, wherein instances of the class are allocated memory space

for the associated field value when the instance of the class is created to support values in preallocated memory space.

23. (Currently Amended) A system as claimed in claim 13 wherein the operation is to set an option value to a second value, the second value having a declared type, and wherein type description is used to checking the operation comprises checking the declared type of a value to be set in a set operation against the type description corresponding to the selected option.

24. (Currently Amended) A system as claimed in claim 13 wherein the operation is one that is to be performed using a selected option value resulting from a get operation, and wherein type description is used to checking the operation comprises checking the legality of an the operation to be performed on a using the selected option value obtained in a get operation.

25. (Currently Amended) A data processing system using an object-oriented programming language comprising:

means for defining compiling a definition of a class which supports an options, each option defined in the class or through a class inheritance hierarchy, and each option having an option value associated therewith data structure having, wherein in instances of the class, references to option values the options are referenceable without allocation of memory space for the full associated option values when the instances of

~~the class are is-created, the option data structure including a type of description of the option values; and~~

wherein compiling the definition comprises

generating a class describing data structure, wherein the class describing data structure is associated with at least one option describing data structure, the at least one option describing data structure comprising a type description corresponding to each option defined in the class or through the class inheritance hierarchy; and

~~compiler-means for compiling an operation on an~~ value associated with a selected option value in an instance of the class, using the type description in the option data structure

wherein compiling the operation comprises searching the at least one option describing data structure associated with the class describing data structure to locate the type description corresponding to the selected option, and using the type description to type check the operation.

26. (Currently Amended) A computer program product comprising:
a computer usable storage medium for storing data; and
a set of computer program instructions ~~using an object-oriented programming language embodied on the computer usable~~ storage medium, including instructions to define-compile a definition of a class which supports an options, each option defined in the class or through a class inheritance hierarchy, and each option having an option value associated therewith ~~data structure having,~~

wherein in instances of the class, references to the options values are
referenceable without allocation of memory space for the full associated option
values when the instances of the class are is created, the option data structure
including a type description of the option values; and

wherein compiling the definition comprises

generating a class describing data structure, wherein the class
describing data structure is associated with at least one option describing
data structure, the at least one option describing data structure comprising
a type description corresponding to each option defined in the class or
through the class inheritance hierarchy; and

compile an operation on the option a value associated with a selected option in
an instance of the class, using the type description of the option value in the option data
structure

wherein compiling the operation comprises searching the at least one option
describing data structure associated with the class describing data structure to locate
the type description corresponding to the selected option, and using the type description
to type check the operation.

27. (Canceled).

28. (Canceled).

29. (Currently Amended) The method of claim 1, wherein the allocation of memory space for an option value in an instance of the class occurs when the option value is set.

30. (Currently Amended) The method of claim 1, wherein the definition of ~~defining the class~~ also supports fields, each field having a field value associated therewith, wherein a data structure having, in instances of the class, ~~references to field values for which memory space is allocated~~ for the associated field values when the instance is created, ~~the data structure including a type description of the field value; and~~ wherein the method further comprises

generating a field describing data structure, the field describing data structure comprising a type description corresponding to each field supported by the class; and

compiling an operation on a field value in an instance of the class, wherein compiling the operation on the associated field value comprises locating the type description corresponding to the field and using the type description ~~in the data structure to type check the operation.~~

31. (New) A method as claimed in claim 1 wherein the at least one option describing data structure further comprises a name corresponding to each option defined in the class or through the class inheritance heirarchy, and wherein compiling the operation further comprises using the name to locate the type description corresponding to the selected option.

32. (New) A method as claimed in claim 31 wherein the at least one option describing data structure comprises an option binding object for each option defined in the class or through the class inheritance hierarchy, wherein each option binding object comprises the name and the type description for the corresponding option.

33. (New) A method as claimed in claim 1 wherein the class describing data structure has a reference to the at least one option describing data structure, and wherein compiling the operation comprises searching the at least one option describing data structure referenced by the class describing data structure to locate the type description corresponding to the selected option, and using the type description to type check the operation.

34. (New) A method as claimed in claim 1 wherein an instance of the first class is associated with a first listing data structure, the first listing data structure having option items comprising option values, the option items being arranged in a first form, and

wherein the method further comprises compiling a definition of a second class which supports options, each option having an option value associated therewith, and wherein an instance of the second class is associated with a second listing data structure, the second listing data structure having option items comprising option values, the option items being arranged in a second form, and

wherein compiling the operation on a value associated with a selected option in an instance of the first class comprises encoding the operation as a method call to an

instance of the first class without regard to form of the listing data structure supported by the first or second class, and

wherein compiling the operation on a value associated with a selected option in an instance of the second class comprises encoding the operation as a method call to an instance of the second class without regard to form of the listing data structure supported by the first or second class.

35. (New) A method as claimed in claim 34 wherein the first form is a linked list and the second form is a hash table.

36. (New) A method as claimed in claim 1 wherein compiling the operation further comprises translating the operation into a method call to the instance of the class.

37. (New) A method as claimed in claim 2 wherein the selected option is defined in a plurality of classes within a class inheritance hierarchy, with a subset of the plurality of classes having change handler code corresponding to the selected option, and further comprising, when the option value associated with the selected option changes, executing the change handler code from the subset of the plurality of classes .

38. (New) A system as claimed in claim 13 wherein the options in an instance of the first class are associated with a first listing data structure, the first listing data

structure having option items comprising option values, the option items associated with the first listing data structure being arranged in a first form, and

wherein the processor is further configured for compiling a definition of a second class which supports options, each option having an option value associated therewith, and wherein the options in an instance of the second class are associated with a second listing data structure, the second listing data structure having option items comprising option values, the option items associated with the second listing data structure being arranged in a second form, and

for compiling the operation on a value associated with a selected option in an instance of the first class further comprises encoding the operation as a method call to an instance of the first class without regard to form of the listing data structure supported by the first or second class.